Angting Cai's Resume

a2cai@ucsd.edu | (+1) 734-877-0657 | https://at1314.github.io/index.html
University of California, La Jolla, CA

EDUCATION

University of California San Diego, *Dept of Computer Science and Engineering* *Sept. 2024 - Dec. 2025*
- Overall GPA: **3.97**/4.0
- Pursuing a Master degree of Computer Science

University of Michigan, *Computer Science and Engineering Department* *Sept. 2022 - May. 2024*
- Overall GPA: **3.88**/4.0
- Honors: Dean's Honor List, 2022

Shanghai Jiaotong University, *UM Joint Institute* *Sept. 2020 - Aug. 2024*
- Overall GPA: 3.53/4.0
- Honors: Undergraduate Excellent Scholarship, 2021

PROFESSIONAL EXPERIENCE

Phoenix Project *May. 2023 - May. 2024*

Research Assistant, Supervised by Prof. Ryan Huang, the CSE Dept., University of Michigan
- Built Phoenix, an OS-supported recovery framework that accelerates high-availability restarts by preserving long-lived process state while discarding transient state.
- Implemented efficient restart support for Glibc/Jemalloc allocator state; worked on custom Linux syscalls, allocator metadata storage, memory-mapping management, and kernel/user-space runtime APIs.
- Developed Python/Shell fault-injection infrastructure to validate Phoenix on real systems such as Redis and LevelDB, repeatedly exercising random-failure recovery paths.
- Co-authored paper accepted at SOSP '25.

Pilot Execution *Dec. 2024 - Mar. 2026*

Research Assistant, Supervised by Prof. Chang Lou, Dept. of Computer Science, University of Virginia
- Studied real-world recovery failures where the recovery action itself can trigger severe, irreversible, or cascading failures in distributed systems.
- Implemented core components of PILOT, an execution model for dry-run simulation of recovery actions in production so operators can observe consequences before commit.
- Applied PILOT to large-scale systems including Hadoop YARN to surface real recovery failures with minimal additional unavailability.
- Co-authored paper accepted at NSDI '26.

Amazon *June. 2025 - Sept. 2025*

Software Developmenet Engineer Intern
- Completed a customer-facing internship project, including making design choices, core implementations, inter/intra-team communications and safe deployment(s) to production.
- Worked with standardized industrial pipeline, including CI-CD deployment, A/B testing and integration tests.
- Worked with Amazon Internal AI tool to enhance the work efficiency when learning new systems and programming languages, coding to implement new features, update/create unit tests and write scripts.

Edge Computer-Use Agent (ECUA) *Sept. 2025 - Dec. 2025*

Systems for LLMs and AI Agents
- Built a fully local computer-use agent on Ubuntu for multi-step GUI and CLI tasks without cloud APIs, using a Perception–Plan–Execute–Reflect loop around Qwen3-VL-8B served with vLLM.
- Implemented hybrid perception with screenshots and accessibility trees, resolution-invariant coordinate rescaling, deterministic grounding, and a constrained action schema to improve execution reliability.
- Added domain-adaptive RAG cheat sheets, loop detection, and reflection mechanisms to mitigate small-model failure modes on app-specific workflows.
- Optimized local inference with FP8, prefix caching, and CUDA graphs, reducing time-to-first-token by ~65% (762 ms → 264 ms); evaluated on 16 OSWorld tasks across 10 domains and achieved 31.25% success, matching a 235B baseline on that subset.

Ultra-Wided Band(UWB) Localization *Dec. 2021 - Apr. 2022*

Student Research Training program, Supervised by Prof. Aimin Tang, SJTU
- Applied multiple DW1000 devices to send signals in 2D/3D indoor space to localize targets by applying methods of

AOA and TOF; Built a system to visualize targets in the map.
- Designed an efficient structure of the system from users' perspective; Created clear and convenient UI on front-end client with QT to visualize data other team members gained from devices.

Pager *Mar. 2023 - Apr. 2023*

Introduction to Operating Systems, Supervised by Prof. Baris Kasikci and Prof. Manos Kapritsos

- Implemented a pager efficiently manages virtual memory for application processes, handling address space creation, read and write faults, address space destruction. Utilized a simulated MMU with a single-level, fixed-size page table and implemented functions such as vm_init, vm_create, vm_fault, vm_destroy and vm_map.
- Built FSM to help implement swap-backed paging and file-backed paging in multi-core scenario

Sharded Key/Value Service with Paxos Groups *Nov. 2023 - Dec. 2023*

Introduction to Distributed Systems, Supervised by Prof. Brian Noble

- With the built Paxos Library and Paxosrsm layer in previous projects, designed and implemented a fault-tolerant key/value storage system that addresses the limitations of a consistent hashing-based system in **Golang**.
- Built and tested a shard master, responsible for managing configurations and assigning shards to replica groups using Paxos, and a sharded key/value server (shardkv) that operates within replica groups.
- Designed and integrated RPCs to secure smooth communication between different servers in the system.

Scalar Intel P6 Style Out-of-Order pipeline *May 2024 - Aug. 2024*

Computer Architecture, Supervised by Prof. Xinfei Guo

- Implemented a Scalar Intel P6 Style Out-of-Order pipeline for VeriSimpleV using Tomasulo Algorithm and ROB (Reorder Buffer) in SystemVerilog.
- Designed and integrated features into the pipeline to realize in-order commit and out-of-order execution of instructions. Improved the performance by integrating new advanced features like Instruction prefetching and Load Store Queue.
- Used Docker to efficiently and consistently run benchmarks to ensure accuracy by comparing the result of the designed OoO pipeline with the baseline, and to find the best designed choices of each module’s detailed structure with best performance.

TEACHING

Instructional Aid for Introduction to Logic Design

- Hold lab, discussion, and office hours weekly for a course on the main B.S. ECE track with 100+ students. Responsible for checking and improving students' hands-on ability in weekly labs.

SKILLS AND INTERESTS

- Research Interests: **Reliable AI/ML systems, LLM inference and agent infrastructure, distributed systems, recovery/fault tolerance, and systems verification**
- Languages: Python, C/C++, Go, Java, JavaScript, Shell
- Systems / AI Infra: Linux, Docker, AWS, Redis, LevelDB, Hadoop YARN, Paxos, vLLM, RAG, CI/CD, fault injection, and inference optimization (FP8, prefix caching, CUDA graphs)